

PROJECT ADMINISTRATION DATA SHEET

ORIGINAL



REVISION NO. _____

Project No. E-24-636DATE: 8/10/81Project Director: Dr. C. M. ShettySchool/~~Lab~~ ISyESponsor: National Science Foundation; Washington, D.C. 20550Type Agreement: Grant No. ECS-8026770 2/28/83Award Period: From 9/1/81 To 8/31/83 (Performance) ----- (Reports)Sponsor Amount: \$34,999

Contracted through:

Cost Sharing: \$1,750 (E-24-350)GTRI/~~GIT~~Title: Optimization with Disjunctive ConstraintsADMINISTRATIVE DATAOCA CONTACT Duane Hutchison x4820

- 1) Sponsor Technical Contact: Abraham H. Haddad, NSF Program Officer; System Theory & Operations Research; Division of Electrical, Computer, & Systems Engineering; Directorate for Engineering and Applied Science; NSF; Washington, D.C. 20550
Tel: 202-357-9618
- 2) Sponsor Admin./Contractual Contact: Hugh L. Lyon, NSF Grants Official; Section II; AAEO/EAS Branch; Division of Grants & Contracts; Directorate for Administration; NSF; Washington, D.C. 20550 Tel: 202-357-9602

Reports: See Deliverable Schedule Security Classification: noneDefense Priority Rating: noneRESTRICTIONSSee Attached NSF Supplemental Information Sheet for Additional Requirements

Travel: Foreign travel must have prior approval - Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of \$500. or 125% of approved proposal budget category.

Equipment: Title vests with GITCOMMENTS: * Includes the usual six (6) month unfunded flexibility period.COPIES TO:

Administrative Coordinator
Research Property Management
Accounting Office

Research Security Services
Reports Coordinator (OCA)
Legal Services (OCA)

EES Research Public Relations
Project File (OCA)
Other:

SPONSORED PROJECT TERMINATION SHEET

Date 5/24/83

Project Title: Optimization with Disjunctive Constraints

Project No: E-24-636

Project Director: Dr. C. M. Shetty

Sponsor: National Science Foundation

Effective Termination Date: 5/12/83

Clearance of Accounting Charges: 5/12/83

Grant/Contract Closeout Actions Remaining:

- ☐ Final Invoice and Closing Documents
- ☒ Final Accounting Report
- ☒ Final Report of Inventions If positive
- ☒ Govt. Property Inventory & Related Certificate
- ☐ Classified Material Certificate
- ☐ Other _____

Assigned to: ISyE (School/Laboratory)

COPIES TO:

Administrative Coordinator
Research Property Management
Accounting
Procurement/EES Supply Services

Research Security Services
~~Reports Coordinator (OCA)~~
Legal Services (OCA)
Library

EES Public Relations (2)
Computer Input
Project File
Other Shetty

Georgia Institute
of
Technology



**INDUSTRIAL and
SYSTEMS ENGINEERING
REPORTS SERIES**

FOR INFORMATION WRITE:

REPORT SERIES LIBRARIAN
SCHOOL OF INDUSTRIAL & SYSTEMS
ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
ATLANTA, GEORGIA 30332

DEVELOPMENT OF VALID INEQUALITIES
FOR DISJUNCTIVE PROGRAMMING

by

B. Ramarao and C. M. Shetty
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

This research was supported by NSF Grant ECS-8026770.

DEVELOPMENT OF VALID INEQUALITIES FOR DISJUNCTIVE PROGRAMMING

B. Ramarao and C. M. Shetty
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

Summary: In this paper we address the question of developing deep cuts for disjunctive programs using rectilinear distance measures. The method is applied to linear complementarity problems where the matrix M need not be copositive plus. Some modifications that are needed as a computational expediency are discussed. The computation results for matrix M of size up to 30×30 are discussed.

1. INTRODUCTION

1. Valid Inequalities for Disjunctive Programming

In this paper we are concerned with nonconvex programs of the type

$$\text{Problem DP:} \quad \text{Minimize } f(x) \quad (1)$$

$$\text{s.t.} \quad x \in X \quad (2)$$

$$x \in \bigcup_{h \in H} S_h \quad (3)$$

where f is continuous, X is a closed subset of the nonnegative orthant of E_n and each S_h is of the form † .

$$\begin{aligned} S_h &= \{x: A^h x > e^h, x > 0\} \quad , h \in H \\ &= \{x: \sum_{j \in N} a_{ij}^h x_j > 1 \text{ for each } i \in Q_h, x > 0\} \quad , h \in H \end{aligned} \quad (4)$$

† Without loss of generality, we have let the right hand side of the constraints to be equal to 1 since we will be dealing with the case where the right hand side is positive.

where e^h is a vector of ones. Such programs have been called Disjunctive Programs and significant theoretical results on the solution of such problems have been presented in the literature. For example, see Balas [1,2,3,4], Glover [6,7], Jeroslow [8,9,10], and Sherali and Shetty [18]. These results were summarized in a monograph by Sherali and Shetty [17], and in this presentation we will be using the notation and concepts presented there.

In the solution of disjunctive programs DP above, a basic approach frequently adopted is to develop a valid inequality (cutting plane) to the disjunction defined by (3).

Definition 1. An inequality $\pi x > \pi_0$ is said to be a valid inequality for the disjunction (3) if

$$x \in S = \bigcup_{h \in H} S_h \text{ implies } \pi x > \pi_0$$

Then, from the pioneering work of Balas, Glover and Jeroslow, the fundamental Disjunctive Cut Principle can be stated as

Theorem 1

Let each S_h be as defined in (4) where H need not be finite. Then, for any choice of nonnegative $\lambda^h = (\lambda_i^h, i \in Q_h)$, the following is a valid inequality for the disjunction (3).

$$\left[\sup_{h \in H} (\lambda^h)^t A^h x \right] > \inf_{h \in H} (\lambda^h)^t e^h$$

(5)

Furthermore, if every system S_h is consistent, then for any valid inequality $\sum_{j \in N} \pi_j x_j > \pi_0$, there exist $\lambda > 0$ such that

$$\pi_0 < \inf (\lambda^h)^t e^h \quad \text{and}$$

$$\sup_{h \in H} \sum_i \lambda_i^h a_{ij} < \pi_j \quad \text{for each } j \in N$$

1.2 Deep Cuts for Disjunctive Programs

In [16], Sherali and Shetty discussed the question of developing "deep cuts" for (3), i.e., how to generate the multipliers λ^h so that the resultant cut is as "deep" as possible. It was shown that maximizing the euclidean or rectilinear distance from the origin to the nonnegative region feasible to the cut would have the desired properties. This distance to the cut $\sum_{j \in N} \pi_j x_j \geq 1$ where $\pi_j > 0$ for some $j \in N$, is given by

$$\begin{aligned} \text{Euclidean Distance:} \quad \theta_e &= 1/\|y\| \text{ where } y_j = \max(0, \pi_j) \\ \text{Rectilinear Distance:} \quad \theta_r &= \frac{1}{\pi_m} \text{ where } \pi_m = \max_{j \in N} \pi_j \end{aligned} \quad (6)$$

For the case where Q_h is a singleton for each $h \in H$, (which we will call Problem DP-1), specification of a deep cut is relatively simple as given by the following theorem [16].

Theorem 2

Consider a disjunctive program DP-1 where

$$S_h = \{x: \sum_{j \in N} a_{1j} x_j \geq 1, x \geq 0\}, h \in H$$

Suppose each S_h is consistent. Let

$$a_{1j}^* = \max_{h \in H} a_{1j}^h, j \in N$$

$$\gamma^h = \text{minimum}_{j: a_{1j}^h > 0} \left\{ \frac{a_{1j}^*}{a_{1j}^h} \right\}, \quad h \in H$$

$$a_{1j}^{**} = \max_{h \in H} a_{1j}^h \gamma^h, \quad j \in N$$

Then the cut

$$\sum a_{1j}^{**} x_j > 1$$

maximizes both criteria in (6) and is a facet of $S = \bigcup_{h \in H} S_h$.

However, for disjunctive programs with $|Q_h| > 1$ (which we will call Problem DP-2), it is not possible to specify the coefficients in explicit form. In this case we need to solve a mathematical program. For maximizing the euclidean distance, the problem to be solved is a quadratic program and in [16] a subgradient optimization scheme is discussed. However, computational results on certain special classes of problems were not too encouraging. This study investigates the algorithmic and computational implications of maximizing the rectilinear distance from the origin to the cut.

Without loss of generality let the multipliers satisfy the condition

$$\sum_{i \in Q_h} \lambda_i^h = 1 \quad \text{and} \quad \lambda_i^h > 0 \quad \text{for each } i \in Q_h \text{ and } h \in H \quad (7)$$

Then noting (5) and (6) the objective function, representing the rectilinear distance from the origin to the cut, to be maximized becomes

$$\text{maximize}_{\lambda} \left[\max_{j \in N} \max_{h \in H} \sum_{i \in Q_h} \lambda_i^h a_{ij}^h \right]^{-1}$$

subject to (7). This multiplier finding problem MF can be written as:

Problem MF. Minimize ξ

s.t. $\xi > \sum_{i \in Q_h} \lambda_i^h a_{ij}^h, \quad h \in H \text{ and } j \in N$

$\sum_{i \in Q_h} \lambda_i^h = 1, \quad h \in H$

$\lambda_i^h > 0, \quad h \in H \text{ and } i \in Q_h$

Note that the constraints are separable in h , and the theorem below shows that Problem MF can be solved by solving Problem MF(h) below for each $h \in H$.

Problem MF(h) Minimize $\{\xi^h: \xi^h > \sum_{i \in Q_h} \lambda_i^h a_{ij}^h \text{ for } j \in N$

$\sum_{i \in Q_h} \lambda_i^h = 1$

$\lambda_i^h > 0 \text{ for } i \in Q_h$

$\xi^h > 0\}$

Theorem 3

Let $\bar{\xi}^h, \bar{\lambda}_i^h$ solve MF(h) for $h \in H$. Then $\bar{\xi} = \max_{h \in H} \bar{\xi}^h$ and $\bar{\lambda}_i^h$ solves Problem MF.

Proof: Follows directly using the Karush-Kuhn-Tucker conditions.

2. APPLICATION TO THE LINEAR COMPLIMENTARITY PROBLEM

2.1. Relaxed Linear Program

In this section we will apply the thoughts developed earlier to the Linear Complementarity Problems (LCP) of the form: Find w and z satisfying

$$\text{Problem LCP:} \quad w - Mz = q \quad (8)$$

$$w, z \geq 0 \quad (9)$$

$$w^t z = 0 \quad (10)$$

where $w, z, q \in E_m$ and M is a $m \times m$ matrix. If M is copositive-plus, Lemke's algorithm [15] can readily be applied. In this paper we will not make this assumption, and in general, M is not copositive-plus. Furthermore, we will assume $q \not\equiv 0$ since otherwise a trivial solution $w = q, z = 0$ is available.

Now, we will introduce an artificial variable z_0 with coefficient +1 in each equation to give a related problem.

$$\text{Problem LP:} \quad \text{Minimize } z_0$$
$$\text{s.t.} \quad w - Mz - ez_0 = q \quad (11)$$

$$w, z, z_0 \geq 0 \quad (12)$$

Letting $z_0 = \max \{-q_i, i = 1, \dots, m\}$, $z = 0$ and $w = q + ez_0$ gives a starting basic feasible solution. The basic thrust of the procedure is to obtain a solution to (11) and (12) with z_0 as nonbasic and with the

complementarity condition satisfied as much as possible. If the complementarity conditions are violated, disjunctive conditions that need be satisfied are developed. Using Problem MF(h) of the previous section we compute the appropriate multipliers to get a deep cut using Theorem 2. A new feasible point is generated and the process repeated.

2.2. Reducing Complementarity Violations

In the implementation, we are faced with the following questions.

- a. If more than two complementary pairs are basic (and positive) in a solution feasible to (8) to (10), which two pairs to pick to develop the cut.
- b. After adding a cut in the iteration towards primal feasibility, if there is a choice in the variable to enter the basis (as is always the case in this instance) which variable to select as the entering variable.

In case (a) above it seems logical that we select the pairs that most contribute to non-complementarity. In case (b) again the same notion holds but taking into account that some of the variables may be negative. The following measure of non-complementarity is therefore used in both instances.

$$C = \sum_{i=1}^m |w_i z_i| \quad (13)$$

Note that when we get a nonnegative solution with $C = 0$ we can terminate with an optimal solution to LCP.

2.3 Development of Cuts

Consider a feasible solution to (8) and (9) with (10) violated. If only one pair (w_k, z_k) is basic with $w_k, z_k > 0$, then the disjunctive statement

$$\text{either } w_k < 0 \text{ or } z_k < 0$$

holds. The expression w_k and z_k in terms of the nonbasic variables is available from the current tableau. These specify the sets S_1 and S_2 with $|H| = 2$ and with exactly one inequality in each set. Theorem 2 yields the deepest cut if each set is consistent.

On the other hand, if two or more pairs are basic we will select two pairs[†] which contribute the most to the measure C defined by (13).

Suppose these are (w_j, z_j) and (w_k, z_k) . The disjunctive statement is $(w, z) \in \bigcup_{h=1}^4 S_h$ where S_i is given below

$$\begin{aligned} S_1: & \quad w_j < 0 \text{ and } w_k < 0 \\ S_2: & \quad w_j < 0 \text{ and } z_k < 0 \\ S_3: & \quad z_j < 0 \text{ and } w_k < 0 \\ S_4: & \quad z_j < 0 \text{ and } z_k < 0 \end{aligned}$$

In this case $|Q_h| = 2$ for each h and $|H| = 4$. Solution of problem MF via the linear programs MF(h) then yields a set of multipliers λ_i^h which reduces it to a disjunctive program DP-1. Theorem 2 then yields the cut if each set is consistent.

[†]Clearly more than two pairs can be selected. But the added computational burden in finding the multipliers does not seem to justify this approach.

2.4. Nonextremal Variables

Observe that the Theorem 2 applies only when the disjunctive statements S_h are consistent. Suppose that (w_k, z_k) is one of the pairs selected for developing a cut and w_k is nonextremal, that is, $w_k > 0$ in any solution to the current set of equations. This is evidenced by nonpositive coefficients in expressing w_k in terms of the nonbasic variables. In this case we must have $z_k = 0$, and hence we pivot out z_k , fix it at the lower bound (zero) and apply the dual simplex method to get back a primal feasible solution.

2.5. Round off Errors and Fathoming.

In some preliminary testing, it was found that even though a point close to the optimal was reached several more iterations were needed to obtain the optimal. This solution was essentially because of round-off errors in the computation of cuts and/or pivoting (even with basis refactorization every 15 iterations). Thus, an attempt will be made to find a complementary solution in the neighborhood. We will call this process fathoming. The first step in fathoming is to verify whether or not we are at a near-complementary solution. This is done by checking whether the following two conditions are met.

- a. At least one variable in the pair (w_i, z_i) is basic for $i = 1, \dots, m$
- b. If both (w_i, z_i) pair are basic, exactly one variable in each pair has a value less than ϵ , a predetermined tolerance. In our program we let $\epsilon = 0.05$.

Suppose the above criteria are satisfied and let

$J = \{j: (w_j, z_j) \text{ basic with either } w_j < \epsilon \text{ or } z_j < \epsilon \text{ but not both}\}.$

We will attempt to find a complementary solution using the following steps.

- (i) For each $j \in J$, replace the variable less than ϵ by a nonbasic slack variable associated with a cut, using the largest (nonzero) element in the corresponding row as the pivot element. We now have a complementary solution since one variable in each pair (w_j, z_j) is nonbasic. Fix each such nonbasic variable at its lower bound (zero).
- (ii) If the solution is infeasible, i.e., some basic variable is negative, apply the dual simplex method with entering variable restricted to the nonbasic slack variables associated with the cuts.

If the solution at the end of step (ii) is feasible, we call the fathoming a success since we have found a solution to the complementary problem. The fathoming can fail in one of three ways.

- Near-complementarity criteria (a) and (b) given above were not satisfied.
- All the variables less than ϵ could not be pivoted out at step (i). In this case we apply the dual simplex method to recover a feasible solution.
- Solution at the end of step (ii) is infeasible. In this case we unblock the variables fixed at zero at the end of step (i) and apply the dual simplex method to recover a feasible solution.

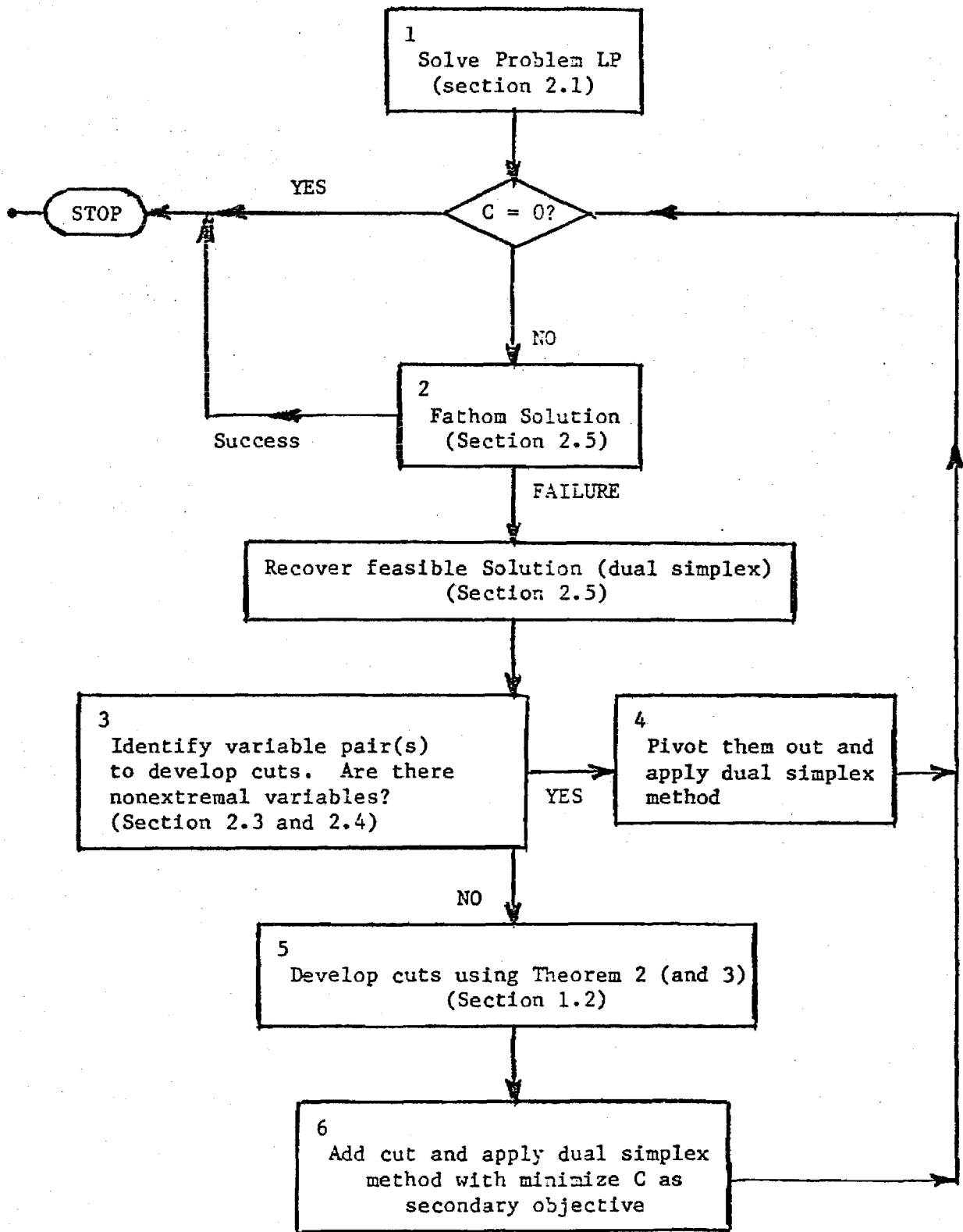


Figure 1. Flow Diagram for solving Problem LCP with M Not Copositive-Plus

2.6. Lemkes' Algorithm

In Section 2.1 we discussed the solution of Problem LP as a linear program to minimize z_0 . Preliminary computational experience showed that the fewer the violations of complementarity, the better was the computational performance. With this in mind, Problem LP was solved using the side-conditions of Lemke [15] as far as possible in selecting an entering variable. This side-condition ensured that complementarity was not violated. Only when no such entering variable was available was the side-condition relaxed.

3.1 Test Problems

To test the algorithm we needed several large test problems with M not copositive-plus. In [11], Kough has given test problems for indefinite Quadratic Programs. Using the Karush-Kuhn-Tucker conditions, one could set it up as a linear complementarity problem. Table 1 gives the pertinent information, and in all three cases, coincidentally, the optimum was reached in Block 1 of the flow diagram even though the M matrix was not copositive-plus. Furthermore, for two of the problems the global maximum was achieved.

Several other test problems were generated by generating random matrices M and q . All of them turned out to have no feasible solution. Existence of a solution could have been ensured by using results of Mangasarian [12,13]; however a specific solution would not be available. Hence, problems with at least one feasible solution was generated as follows.

Table 1. Results of the Three Indefinite Quadratic Problems (Kough [11])

Problem No.	(# of variables, # of constraints)	# of Lemke pivots	Solution	Computation time in cpu seconds
1	(4,5)	15	x = (12.9924, 1.8515) y = (11.9968, 3.3143)	0.834
2	(4,8)	14	x = (17.4623, 14.7453) * y = (13.5071, 14.0164)	1.029
3	(4,8)	10	x = (16.5911, 0.0) * y = (27.8055, 11.9603)	0.639

* Global optimum

- (i) The $m \times m$ M -matrix was generated columnwise by generating m^2 uniform random integer variables in the range -50 to $+50$. The IMSL subroutine GGUBFS was used for this purpose. The elements of the major diagonal, in general, will have some negative elements. In the case this is not so, two columns were interchanged to give at least one negative element. This will ensure that the matrix is not copositive-plus.
- (ii) A number p was arbitrarily set equal to $m/5$. A solution vector \bar{z} was generated with first p elements equal to zero and the remaining $(m-p)$ elements equal to an integer random number between 2 and 5 (inclusive)
- (iii) Partitioning M as

$$M = \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array} \right]$$

where M_{11} is $p \times p$, the \bar{w} solution vector was generated by

$$\bar{w}_i = \begin{cases} 2 & \text{if } [M_{11}|M_{12}]\bar{z} > 0 \\ 0 & \text{if } [M_{11}|M_{12}]\bar{z} = 0 \\ 1 & \text{if } [M_{11}|M_{12}]\bar{z} < 0 \end{cases} \quad \text{for } i = 1, \dots, p$$

and $\bar{w}_i = 0$ for $i = p+1, \dots, m$.

- (iv) The vector q is then uniquely given by $q = \bar{w} - M\bar{z}$.

Note that q_i could be negative (and, in general, it will be). Also note that the solution (\bar{w}, \bar{z}) is complementary.

3.2 Computers Used and Software

A CDC CYBER 170/730 mainframe operating under the NOS 1.4 operating system was used. Because of the flexibility offered, the XMP package of subroutines for Linear Programming by Marsten [14] were used. In particular, the flexibility of fixing variables was useful in blocks 2 and 4.

3.3 Computational Results

Ten each of 20x20 and 30x30 problems were generated as in Section 3.1 above. They are identified by PB-20-01 to PB-20-10 and PB-30-01 to PB-30-10 respectively in subsequent discussions. The essential characteristics of these problems is given in Table 2.

Table 3 summarizes the computational results. The average time that the algorithm took to solve a problem of size $m = 20$ was 20.259 c.p.u. seconds. For a problem of size 30, two of the problems could not be solved within the limits of 60 allowable cuts for the program. The average computation time for the remaining 8 30x30 problems was 56.16 c.p.u. seconds.

Note that the fathoming procedure was used infrequently, but when used it yielded the solution in all, except one, instance. Also in block 3, no nonextremal variable was identified in all cases.

In column 6 of Table 3 we have presented a measure called Set Difference. If $\hat{x} = (\hat{w}, \hat{z})$ is the solution at the end of block 1 of the flow diagram, and $\bar{x} = (\bar{w}, \bar{z})$ is the solution when the algorithm terminated, the set difference S is given by

Table 2. Problem Data

Problem No.	# of Negative Main Diagonal Elements	# of Negative Elements in M Matrix	Density of M	# of Negative Elements in q Vector
PB-20-01	9	202	0.99	11
20-02	11	188	0.9775	12
20-03	11	188	0.9775	15
20-04	8	191	0.97	13
20-05	7	204	0.975	8
20-06	13	200	0.9875	9
20-07	11	184	0.975	14
20-08	10	200	0.975	10
20-09	11	184	0.9750	14
20-10	8	198	0.9825	9
PB-30-01	13	452	0.98	12
30-02	17	436	0.9744	11
30-03	10	421	0.9789	14
30-04	14	440	0.9822	13
30-05	16	428	0.9833	15
30-06	20	442	0.9767	18
30-07	16	460	0.9867	11
30-08	16	439	0.9811	13
30-09	13	438	0.9789	16
30-10	16	460	0.9867	11

Table 3. Summary of Results

Problem No.	Computation time in cpu seconds	Total # of cuts (cuts based on single pairs)	# of times fathoming used	# of nonextremal variables removed	Set Difference S	# of Lemke pivots
PB-20-01	19.76	6(0)	1*	0	16	3
20-02	27.57	8(0)	1*	0	12	8
20-03	26.97	10(3)	1*	0	11	57
20-04	19.44	5(2)	1*	0	8	95
20-05	25.85	6(1)	0	0	6	115
20-06	14.345	2(0)	0	0	9	125
20-07	18.616	6(3)	1*	0	17	9
20-08	12.233	3(1)	0	0	11	54
20-09	17.972	6(3)	1*	0	17	9
20-10	20.016	6(1)	0	0	18	5
Average for 20x20	20.259	5.60	0.60	0	12.5	48.3
PB-30-01	25.21	4(1)	1	0	16	16
30-02	---	---	-	-	-	---
30-03	94.115	19(1)	0	0	19	6
30-04	28.604	4(0)	0	0	15	44
30-05	37.084	6(1)	1*	0	20	50
30-06	113.684	16(2)	0	0	22	64
30-07	61.28	10(0)	0	0	15	27
30-08	32.734	0(0)	0	0	0	376
30-09††	---	---	-	-	---	---
30-10	56.589	10(1)	0	0	13	27
Average for 30x30	56.163	9.86	0.286	0	17.14	76.25

* indicates solution found upon fathoming
 †† not solved within program limits (60 cuts)

Table 4. Computational Results: Cuts Based on (a) Single Pair
and (b) Two Pairs of Violations

Problem No.	Computation Time in cpu seconds		# of Cuts Added	
	Single Pair†	Two Pairs††	Single Pair†	Two Pairs††
PB-20-01	10.488	19.760	4	6
20-02	35.260	27.57	17	8
20-03	> 140	26.97	--	10
20-04	> 180.00	19.44	> 70	5
20-05	16.193	25.85	4	6
20-06	> 140	14.345	--	2
20-07	27.408	18.616	13	6
20-08	18.857	12.233	10	3
30-02	> 140	> 140	--	--
30-04	45.488	28.604	10	4

† On CDC CYBER 170/760

†† On CDC CYBER 170/730

$$S = \{j: x_j \text{ is basic in } \hat{x} \text{ but not in } \bar{x}\} \cup \{j: x_j \text{ is basic in } \bar{x} \text{ but not in } \hat{x}\}$$

This was computed because it was suspected that an algorithm (like Lemke's) which is likely to yield a solution at the end of block 1 closer to the optimal would have a significant advantage. Table 3 does not bear out this conjecture strongly. Another interesting fact worth mentioning is that in all cases, except Problem PB-30-08, Lemke's algorithm in block 1 ended with a ray termination.

The following problem from [5] was of interest because two columns in the matrix were dependent, the LCP Solutions were degenerate and infinite. An interesting fact about this problem is that the algorithm demonstrated the presence of nonexternal variables and found the solution $w = (2,0,0,0)$ and $z = (0,0,4,1)$.

It may be recalled a measure C defined in (13) of section 2 was used to drive the algorithm towards a complementary solution. In general C

$$m = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad q = \begin{bmatrix} 1 \\ -1 \\ 3 \\ 1 \end{bmatrix}$$

will decrease at each iteration except possibly when a cut is added. In 58 instances of cuts for 20x20 problems, only in 9 instances C increased marginally.

3.4 Alternatives in Developing Cuts

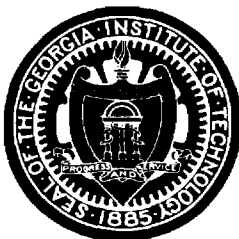
In Block 3, two pairs of complementary notations was used whenever available. In each such case, 4 linear subproblems had to be solved to develop a cut. Generation of data to solve these linear programs contributed significantly to the total solution time. An alternative would be to always select a complementary pair, both positive, to generate a cut. Table 4 compares this strategy with the former. The solution times when two pairs were used were on a CDC CYBER 170/730 whereas the solution times when a single pair was used were on a CDC CYBER 170/760 (which is two to three times faster). It clearly shows the superiority of the proposed method to develop deep cuts.

References

1. Balas, E., "Intersection Cuts from Disjunctive Constraints," Management Science Research Report, No. 330, Carnegie-Mellon University, February 1974.
2. Balas, E., "Disjunctive Programming: Properties of the Convex Hull of Feasible Points," Management Science Research Report 348, GSIA, Carnegie-Mellon University, 1974.
3. Balas, E., "Disjunctive Programming: Cutting Planes from Logical Conditions," Nonlinear Programming, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1975.
4. Balas, E., "Disjunctive Programming," Management Science Research Report No. 415, Carnegie-Mellon University, August 1977.
5. Cottle, R. W. and G. B. Dantzig, "Complementary Pivot Theory of Mathematical Programming," Linear Algebra and Its Applications, 1, 103-125, 1968.
6. Glover, F., "Polyhedral Convexity Cuts and Negative Edge Extensions," Zeitschrift fur Operations Research, 18, pp. 181-186, 1974.
7. Glover, F., "Polyhedral Annexation in Mixed Integer and Combinatorial Programming," Mathematical Programming, Vol. 8, pp. 161-188, 1975. (See also MSRS Report 73-9, University of Colorado, August 1973.)
8. Jeroslow, R. G., "The Principles of Cutting Plane Theory: Part I", (with an addendum), GSIA, Carnegie-Mellon University, February 1974.
9. Jeroslow, R. G., "Cutting Planes for Complementarity Constraints," SIAM Journal of Control and Optimization, Vol. 16, No. 1, pp. 56-62, 1976.
10. Jeroslow, R. G., "Cutting Plane Theory: Disjunctive Methods," Annals of Discrete Mathematics, Vol. 1, pp. 293-330, 1977.
11. Kough, P. F., "The Indefinite Quadratic Programming Problem," Operations Research, 27, 516-533, 1979.
12. Mangarasian, O., "Characterization of Linear Complementary Problems," Mathematical Programming Study, 7, 74-87, 1978.
13. Mangasarian, O., "Characterization of Bounded Solutions of Linear Complementarity Problems," Mathematical Programming Study, 19, 153-166, 1982.
14. Marsten, R. E., Users Guide for XMP, University of Arizona, Tuscon developed under NSF Grant MCS-76-01311 and MCS-76-01311 A01, 1978.
15. Lemke, C. E., "On Complementary Pivot Theory," Mathematics of the Decision Sciences, (G. B. Dantzig and A. F. Veinott, Eds.), 1968.

16. Sherali, H. D. and C. M. Shetty, "Deep Cuts in Disjunctive Programming," Naval Research Logistics Quarterly, Vol. 27, pp. 453-475, 1980.
17. Sherali, H. D. and C. M. Shetty, Optimization with Disjunctive Constraints, Lecture notes in Economics and Mathematical Systems #181, Springer-Verlag, 1980.
18. Sherali, H. D. and C. M. Shetty, "Disjunctive Programming, Polyhedral Annexation Techniques, and Nondominated Disjunctive Cutting Planes," Georgia Institute of Technology, 1979.

**SCHOOL OF INDUSTRIAL
AND
SYSTEMS ENGINEERING**



**GEORGIA INSTITUTE
OF TECHNOLOGY
ATLANTA, GEORGIA 30332**

PLEASE READ INSTRUCTIONS ON REVERSE BEFORE COMPLETING

PART I-PROJECT IDENTIFICATION INFORMATION

Institution and Address Georgia Institute of Technology Atlanta, Georgia 30332	2. NSF Program Systems Theory and Operations Research	3. NSF Award Number ECS-80-26770
	4. Award Period From 9/1/81 To 2/28/83	5. Cumulative Award Amount \$34,999


Project Title

Optimization with Disjunctive Constraints

PART II-SUMMARY OF COMPLETED PROJECT (FOR PUBLIC USE)

The research shows how one may obtain using linear programming the coefficients needed to aggregate constraints for specifying deep cuts in the context of disjunctive programs. It is shown that the linear program can be decomposed into several smaller linear programs. The methodology is applied to linear complementarity problems of size up to 30×30 whose defining matrix M is not copositive-plus for which currently there is no solution procedure. Several refinements are introduced, including the use of Lemke's procedure to give a better starting point and the use of a secondary objective function after each cut is generated. It is shown that use of two pairs of complementary variables which contribute most to the violation of the complementarity condition is superior to obtaining cuts via a single complementary pair, both positive. A scheme, referred to as fathoming, was used to verify whether a complementary solution exists in the neighborhood of the current point. This scheme was very effective in reducing the computation time whenever round-off errors caused the procedure to hover around the complementary solution.

PART III-TECHNICAL INFORMATION (FOR PROGRAM MANAGEMENT USES)

ITEM (Check appropriate blocks)	NONE	ATTACHED	PREVIOUSLY FURNISHED	TO BE FURNISHED SEPARATELY TO PROGRAM	
				Check (✓)	Approx. Date
Abstracts of Theses	X				
Publication Citations		X			
Data on Scientific Collaborators		X			
Information on Inventions	X				
Technical Description of Project and Results		X			
Other (specify) Copy of research report		X			
Principal Investigator/Project Director Name (Typed) C. M. Shetty	3. Principal Investigator/Project Director Signature 			4. Date 5/8/83	

Publications and Presentation

1. "Development of Valid Inequalities for Disjunctive Programming," Research Report J-82-10, October 1982, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332. Also presented at the Joint ORSA/TIMS Conference, San Diego, October 25-27, 1982.
2. A paper based on the research is currently being reviewed for publication in the Naval Research Logistics Quarterly.

List of Collaborators

1. Mr. B. Ramarao, Ph.D. student, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332

OPTIMIZATION WITH DISJUNCTIVE CONSTRAINTS

C. M. Shetty

Principal Investigator

NSF Research Grant ECS-8026770

Final Technical Report

1. Introduction

The research covered by this final report deals with the development of effective cuts for disjunctive programs, and its computational testing. The research was funded by NSF Grant ECS-8026770. The specific objectives of the research as proposed were

1. To develop an alternative scheme for obtaining the coefficients of deep nondominated cuts for general disjunctive programs.
2. To adapt and implement the above scheme for specialized problems.

An earlier research grant (ENG-77-23683) covered the theoretical aspects of the development of valid cutting planes and finitely convergent algorithms. The present research focuses on development of methodology to obtain deep cuts using linear programming and to test the methodology on linear complementarity problems. The research results have been widely distributed under Report Series J-82-10, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332 and was presented at the Joint ORSA/TIMS Conference at San Diego, October 25-27, 1982. A paper on the subject is currently under the refereeing process.

2. The Disjunctive Program

The disjunctive program (DP) we will be dealing with is a nonconvex program of the type

$$\begin{array}{lll} \text{Problem DP:} & \text{Minimize } f(x) & (1) \\ & \text{s.t. } x \in X & (2) \\ & x \in \bigcup_{h \in H} S_h & (3) \end{array}$$

where f is continuous, X is a closed subset of the nonnegative orthant of E_n and each S_h is of the form

$$S_h = \{x: A^h x \geq e^h, x \geq 0\}, h \in H \quad (4)$$

$$= \{x: \sum_{j \in N} a_{ij}^h x_j \geq 1 \text{ for each } i \in Q_h, x \geq 0\}, h \in H$$

where e^h is a vector of ones. The reader may observe that this last assumption is not restrictive in that all problems with positive right hand sides in defining S_h can be reduced to this form. Significant theoretical results on the solution of disjunctive problems have been presented in the literature. For example, see Balas [1,2,3,4], Glover [6,7], Jeroslow [8,9,10], and Sherali and Shetty [18]. These results were summarized in a monograph by Sherali and Shetty [17], and in this presentation we will be using the notation and concepts presented there.

In the solution of disjunctive programs DP above, a basic approach frequently adopted is to develop a valid inequality (cutting plane) to the disjunction defined by (3).

Definition 1. An inequality $\pi x \geq \pi_0$ is said to be a valid inequality for the disjunction (3) if

$$x \in S = \bigcup_{h \in H} S_h \text{ implies } \pi x \geq \pi_0$$

Then, from the pioneering pioneering work of Balas, Glover and Jeroslow, the fundamental Disjunctive Cut Principle can be stated as

Theorem 1

Let each S_h be as defined in (4) where H need not be finite. Then, for any choice of nonnegative $\lambda^h = (\lambda_i^h, i \in Q_h)$, the following is a valid inequality for the disjunction (3).

$$\sup_{h \in H} (\lambda^h)^t A^h x > \inf_{h \in H} (\lambda^h)^t e^h \quad (5)$$

Furthermore, if every system S_h is consistent, then for any valid inequality $\sum_{j \in N} \pi_j x_j > \pi_0$, there exist $\lambda > 0$ such that

$$\pi_0 < \inf_{h \in H} (\lambda^h)^t e^h \quad \text{and}$$

$$\sup_{h \in H} \sum_i \lambda_i^h a_{ij} < \pi_j \quad \text{for each } j \in N$$

3. Deep Cuts Using Linear Programming

In [16], Sherali and Shetty discussed the question of developing "deep cuts" for (3), i.e., how to generate the multipliers λ^h so that the resultant cut is as "deep" as possible. It was shown that maximizing the euclidean or rectilinear distance from the origin to the nonnegative region feasible to the cut would have the desired properties. This distance to the cut $\sum_{j \in N} \pi_j x_j > 1$ where $\pi_j > 0$ for some $j \in N$, is given by

$$\begin{aligned} \text{Euclidean Distance:} \quad \theta_e &= 1/\|y\| \text{ where } y_j = \max(0, \pi_j) \\ \text{Rectilinear Distance:} \quad \theta_r &= \frac{1}{\pi_m} \text{ where } \pi_m = \max_{j \in N} \pi_j \end{aligned} \quad (6)$$

For the case where Q_h is a singleton for each $h \in H$, (which we will call Problem DP-1), specification of a deep cut is relatively simple as given by the following theorem [16].

Theorem 2

Consider a disjunctive program DP-1 where

$$S_h = \{x: \sum_{j \in N} a_{1j}^h x_j > 1, x > 0\}, h \in H$$

Suppose each S_h is consistent. Let

$$a_{1j}^* = \max_{h \in H} a_{1j}^h, j \in N$$

$$\gamma^h = \min_{j: a_{1j}^h > 0} \left\{ \frac{a_{1j}^*}{a_{1j}^h} \right\}, h \in H$$

$$a_{1j}^{**} = \max_{h \in H} a_{1j}^h \gamma^h, j \in N$$

Then the cut

$$\sum a_{1j}^{**} x_j > 1$$

maximizes both criteria in (6) and is a facet of $S = \bigcup_{h \in H} S_h$.

However, for disjunctive programs with $|Q_h| > 1$ (which we will call Problem DP-2), it is not possible to specify the coefficients in explicit form. In this case we need to solve a mathematical program. For maximizing the euclidean distance, the problem to be solved is a quadratic program and in [16] a subgradient optimization scheme is discussed. However, computational results on certain special classes of

problems were not too encouraging. This study investigates the algorithmic and computational implications of maximizing the rectilinear distance from the origin to the cut.

Without loss of generality let the multipliers satisfy the condition

$$\sum_{i \in Q_h} \lambda_i^h = 1 \quad \text{and} \quad \lambda_i^h > 0 \quad \text{for each } i \in Q_h \text{ and } h \in H \quad (7)$$

Then noting (5) and (6) the objective function, representing the rectilinear distance from the origin to the cut, to be maximized becomes

$$\underset{\lambda}{\text{maximize}} \quad \left[\max_{j \in N} \max_{h \in H} \sum_{i \in Q_h} \lambda_i^h a_{ij}^h \right]^{-1}$$

subject to (7). This multiplier finding problem MF can be written as:

Problem MF. Minimize ε

subject to $\varepsilon > \sum_{i \in Q_h} \lambda_i^h a_{ij}^h, \quad h \in H \text{ and } j \in N$

$\sum_{i \in Q_h} \lambda_i^h = 1, \quad h \in H$

$\lambda_i^h > 0, \quad h \in H \text{ and } i \in Q_h$

Note that the constraints are separable in h , and Theorem 3 below shows that Problem MF can be solved by solving Problem MF(h) for each $h \in H$.

Problem MF(h) Minimize ε^h

subject to $\varepsilon^h > \sum_{i \in Q_h} \lambda_i^h a_{ij}^h \text{ for } j \in N \quad (8)$

$$\sum_{i \in Q_h} \lambda_i^h = 1 \quad (9)$$

$$\lambda_i^h > 0 \text{ for } i \in Q_h \quad (10)$$

Theorem 3

Let $\bar{\epsilon}^h, \bar{\lambda}_i^h$ solve MF(h) for $h \in H$. Then $\bar{\epsilon} = \max_{h \in H} \bar{\epsilon}^h$ and $\bar{\lambda}_i^h$ solves

Problem MF.

Proof: Note that $\bar{\epsilon}$ and $\bar{\lambda}_i^h$ are feasible to problem MF. Let

$\hat{h} = \arg \max_{h \in H} \bar{\epsilon}^h$ so that $\bar{\epsilon} = \bar{\epsilon}^{\hat{h}}$. For each $h \in H$, let the optimal dual

variables associated with (8), (9) and (10) respectively be $\pi_j^h, j \in N, \pi_0^h$ and $y_i^h, i \in Q^h$. Now, let

$$\bar{\pi}_j^h = \begin{cases} \pi_j^h & \text{for } h = \hat{h} \\ 0 & \text{otherwise} \end{cases}$$

$$\bar{\pi}_0^h = \begin{cases} \pi_0^h & \text{for } h = \hat{h} \\ 0 & \text{otherwise} \end{cases}$$

$$\bar{y}_i^h = \begin{cases} y_i^h & \text{for } h = \hat{h} \\ 0 & \text{otherwise} \end{cases}$$

Since π_j^h, π_0^h and y_i^h satisfy the Karush-Kuhn-Tucker conditions, we have for each $h \in H$

$$\sum_{j=1}^n \pi_j^h = 1$$

$$\sum_{j=1}^n \pi_j^h a_{ij}^h + \pi_0^h - y_i^h = 0 \quad i \in Q^h$$

$$\pi_j^h (\sum_{i \in Q^h} a_{ij}^h \bar{\lambda}_i^h - \bar{\xi}^h) = 0 \quad j \in N$$

$$y_i^h \bar{\lambda}_i^h = 0 \quad i \in Q^h$$

$$y_i^h, \pi_j^h > 0$$

From the above conditions and noting the definitions of $\bar{\pi}_j^h$, $\bar{\pi}_0^h$ and \bar{y}_i^h , we have

$$\sum_{h \in H} \sum_{j=1}^n \bar{\pi}_j^h = 1$$

$$\sum_{j=1}^n \bar{\pi}_j^h a_{ij}^h + \bar{\pi}_0^h - \bar{y}_i^h = 0 \quad i \in Q^h, \quad h \in H$$

$$\bar{\pi}_j^h (\sum_{i \in Q^h} a_{ij}^h \bar{\lambda}_i^h - \bar{\xi}^h) = 0 \quad j \in N, \quad h \in H$$

$$\bar{y}_i^h \bar{\lambda}_i^h = 0 \quad i \in Q^h, \quad h \in H$$

$$\bar{y}_i^h, \bar{\pi}_j^h > 0 \quad i \in Q^h, \quad j \in N \quad h \in H$$

These are precisely the Karush-Kuhn-Tucker conditions for Problem MF, and the theorem is proved.

4. Strategy for Solving Linear Complementarity Problems

In this section we will apply the thoughts developed earlier to the Linear Complementarity Problems (LCP) of the form: Find w and z satisfying

$$\text{Problem LCP:} \quad w - Mz = q \quad (11)$$

$$w, z \geq 0 \quad (12)$$

$$w^t z = 0 \quad (13)$$

where $w, z, q \in E_m$ and M is a $m \times m$ matrix. If M is copositive-plus, Lemke's algorithm [15] can readily be applied. In this paper we will not make this assumption, and in general, M is not copositive-plus.

Furthermore, we will assume $q \neq 0$ since otherwise a trivial solution $w = q, z = 0$ is available.

Now, we will introduce an artificial variable z_0 with coefficient -1 in each equation to give a related problem.

$$\begin{aligned} \text{Problem LP:} \quad & \text{Minimize } z_0 \\ & \text{s.t.} \quad w - Mz - ez_0 = q \\ & \quad \quad w, z, z_0 \geq 0 \end{aligned}$$

Letting $z_0 = \max \{-q_i, i = 1, \dots, m\}$, $z = 0$ and $w = q + ez_0$ gives a starting basic feasible solution. The basic thrust of the procedure is to obtain a solution to Problem LP (with some "side conditions" discussed later). If it does not solve LCP, then cuts are used based on the disjunctive principle.

5. Development of Cuts

Note that $z_0 = \max \{-q_i, i = 1, \dots, m\}$, $z = 0$ and $w = q + ez_0$ gives a starting basic feasible solution to Problem LP. A primitive algorithm tested was as follows:

Solve LP as a linear program starting from the above solution. If the optimum solution has z_0 nonbasic and $w^t z = 0$, stop; otherwise

let $w_k z_k = \max_i w_i z_i > 0$. Using Theorem 2 derive a cut as discussed later in the section. The procedure is repeated after applying the dual simplex method.

The above procedure was used to solve Kough's indefinite quadratic problems [11] with little success within the time limite of 60 seconds.

As a modification of the above algorithm, instead of using only one violated pair of variables, whenever possible two pairs (w_j, z_j) , (w_k, z_k) violating complementarity were used. The selection criterion adopted was to use those indices giving the largest and second largest $w_i z_i$ values. As seen below, this leads to the solution of four problems MF(h) to obtain the multiplier values. Theorem 2 is then used to defined the cut. The current point is then infeasible. The procedure was repeated after using the dual simplex method. This method, although better than the primitive algorithm presented earlier, did not perform well on randomly generated problems.

We explain below how in each of the cases discussed above violations of complementarity is used to define disjunctive constraints. Consider a feasible solution to (11) and (12), with (13) violated. If each set is only one pair (w_k, z_k) is basic with $w_k, z_k > 0$, then the disjunctive statement

$$\text{either } w_k \leq 0 \text{ or } z_k \leq 0$$

holds. The expression w_k and z_k in terms of the nonbasic variables is available from the current tableau. These specify the sets S_1 and S_2 with $|H| = 2$ and with exactly one inequality in each set. Theorem 2 yields the deepest cut if each set is consistent.

On the other hand, if two or more pairs are basic we will select two

pairs[†] which contribute the most to the violation of complementarity as

measured by $C = \sum_{i=1}^n |w_i z_i|$. This measure takes into account that

currently some of the variable values may be negative. Suppose these

are (w_j, z_j) and (w_k, z_k) . The disjunctive statement is $(w, z) \in \bigcup_{h=1}^4 S_h$ where

S_1 is given below

$$S_1: \quad w_j < 0 \text{ and } w_k < 0$$

$$S_2: \quad w_j < 0 \text{ and } z_k < 0$$

$$S_3: \quad z_j < 0 \text{ and } w_k < 0$$

$$S_4: \quad z_j < 0 \text{ and } z_k < 0$$

In this case $|Q_h| = 2$ for each h and $|H| = 4$. Solution of problem MF

via the linear programs MF(h) then yields a set of multipliers λ_i^h

which reduces it to a disjunctive program DP-1. Theorem 2 then yields

the cut each set is consistent.

6. Nonextremal Variables

Recall that even with cuts generated based on two violations of complementarity, a feasible solution to Problem LCP was not found within the time limit specified. It was conjectured that this may be on account of the presence of nonextremal variables, that is, variables that are positive in any solution to the current set of equations.

Observe that the Theorem 2 applies only when the disjunctive statements S_h are consistent. Suppose that (w_k, z_k) is one of the pairs selected for developing a cut and w_k is nonextremal, that is,

[†]Clearly more than two pairs can be selected. But the added computational burden in finding the multipliers does not seem to justify this approach.

$w_k > 0$ in any solution to the current set of equations. This is evidenced by nonpositive coefficients in expressing w_k in terms of the nonbasic variables. In this case we must have $z_k = 0$, and hence we pivot out z_k , fix it at the lower bound (zero) and apply the dual simplex method to get back a primal feasible solution. The subroutine to perform this was added to the program along with other modifications discussed below. It turned out later that this subroutine was never called, indicating that nonextremal variables was never a problem in the problems tested.

7. Lemkes' Algorithm

Attention was then focussed on the number of complementarity violations in the optimal tableau to problem LP. If this was large, it seems reasonable to assume that a large number of cuts that would be needed to solve LCP. Preliminary computational experience showed that the fewer the violations of complementarity, the better was the computational performance. With this in mind, Problem LP was solved using the side-conditions of Lemke [15] as far as possible in selecting an entering variable. This side-condition ensured that complementarity was not violated. Only when no such entering variable was available and if complementarity was still violated, was the side-condition relaxed i.e., starting from this point, problem LP was solved as a linear program. Although this modification was reasonably useful, it did not result in any significant reduction in solution times. This feature was retained in the final program.

8. Secondary Objective Function

Although the procedure in Section 7 did take into account the violation of complementary on the beginning, it did not use this type of information once problem LP was solved. At this stage z_0 is nonbasic, and during the dual simplex phase the objective function $\text{Min } z_0$ was not helpful in moving toward a complementary solution since all reduced cost coefficients are zero. A procedure is needed in the dual simplex phase that chooses a variable to enter the basis which reduces total complementary infeasibility while maintaining z_0 as a nonbasic variable. This can be accomplished by

- a. Retaining z_0 at its lower bound and not including it in the set of candidate variables to enter.
- b. Using a secondary objective function given below which reflects total complementary infeasibility and takes into account that some of the variables may be negative

$$C = \sum_{i=1}^m |w_i z_i| \quad (14)$$

This measure C provided a significant improvement in reducing the number of cuts added, thereby accelerating convergence. Note that when we get a nonnegative solution with $C = 0$ we can terminate with an optimal solution to LCP.

9. Round off Errors and Fathoming.

In some preliminary testing, it was found that even though a point close to the optimal was reached several more iterations were needed to obtain the optimal and the same variables were taking on small values

relative to the other variable values. This was essentially because of round-off errors in the computation of cuts and/or pivoting (even with basis refactorization every 15 iterations). Thus, an attempt was made to find a complementary solution in the neighborhood. We will call this process fathoming. The first step in fathoming is to verify whether or not we are at a near-complementary solution. This is done by checking whether the following two conditions are met.

- a. At least one variable in the pair (w_i, z_i) is basic for $i = 1, \dots, m$
- b. If both (w_i, z_i) pair are basic, exactly one variable in each pair has a value less than ϵ , a predetermined tolerance. In our program we let $\epsilon = 0.05$.

Suppose the above criteria are satisfied and let

$J = \{j: (w_j, z_j) \text{ basic with } \underline{\text{either}} w_j < \epsilon \text{ or } z_j < \epsilon \text{ but not both}\}.$

We will attempt to find a complementary solution using the following steps.

- (i) For each $j \in J$, replace the variable less than ϵ by a nonbasic slack variable associated with a cut, using the largest (nonzero) element in the corresponding row as the pivot element. We now have a complementary solution since one variable in each pair (w_j, z_j) is nonbasic. Fix each such nonbasic variable at its lower bound (zero).
- (ii) If the solution is infeasible, i.e., some basic variable is negative, apply the dual simplex method with entering variable

restricted to the nonbasic slack variables associated with the cuts.

The fathoming can fail in one of three ways.

- Near-complementarity criteria (a) and (b) given above were not satisfied.
- All the variables less than ϵ could not be pivoted out at step (i). In this case we apply the dual simplex method to recover a feasible solution.
- Solution at the end of step (ii) is infeasible. In this case we unblock the variables fixed at zero at the end of step (i) and apply the dual simplex method to recover a feasible solution.

If the solution at the end of step (ii) is feasible, we call the fathoming a success since we have found a solution to the complementary problem. Implementation of the fathoming procedure was highly successful. Out of the 30 problems where the fathoming procedure was invoked, in 29 cases the complementary solution was found in the fathoming step. The flow diagram for the algorithm tested is depicted in Figure 1.

10. Computer Used and Software

A CDC CYBER 170/730 mainframe operating under the NOS 1.4 operating system was used. Because of the flexibility offered, the XMP package of subroutines for Linear Programming by Marsten [14] were used. The package was developed under NSF Grants MCS-76-01311 and MCS-76-01311-A01. The particular package used was last modified on February 15, 1979 and required considerable debugging before it was fully operational.† XMP is

†We understand that the authors have since corrected and improved the package.

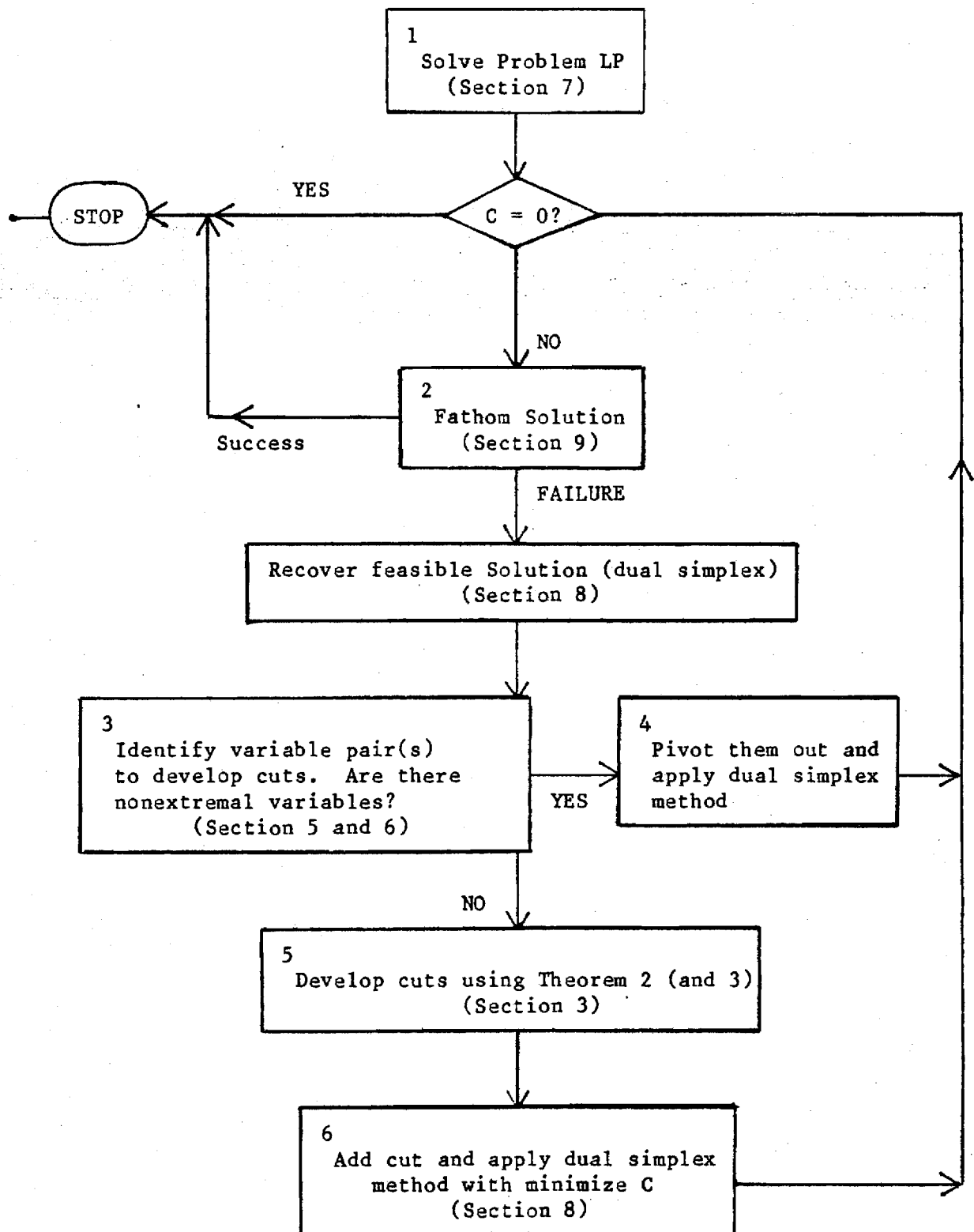


Figure 1. Flow Diagram for solving Problem LCP with M Not Copositive-Plus

a structured user oriented library of subroutines used to solve linear programs. The versatility of the library proved to be very useful in its application to the problem at hand. In particular the features most helpful were:

1. The ability of selecting a pivot-element based on criteria other than those built in to the Primal Simplex and Dual Simplex algorithms. This facilitated the ability to build in alternative solution schemes, such as Lemke's procedure and also allowed the handling of the secondary objective.
2. The ability of locking variables either into the basis or out of the basis. This feature was used in handling of non-extremal variables and in the subroutine to find a complementary solution when the complementary infeasibility was less than a prescribed tolerance, i.e., in the fathoming step.

The XMP package had one major drawback in implementing cutting plane procedure, that is, rows had to be reserved for the number of cuts that were added. This meant that we have to estimate of the number of cuts needed by a particular problem or the space reserved could be as large as the available memory would allow. More importantly, each such reserved row had to have specific entries, e.g., zeros. This meant that each column of the matrix input had a large number of zeros

11. Test Problems

To test the algorithm we needed several large test problems with M not copositive-plus. In [11], Kough has given test problems for indefinite

Quadratic Programs. Using the Karush-Kuhn-Tucker conditions, one could set it up as a linear complementarity problem. Table 1 gives the pertinent information, and in all three cases, coincidentally, the optimum was reached in Block 1 of the flow diagram even though the M matrix was not copositive-plus. Furthermore, for two of the problems the global maximum was achieved.

Several other test problems were generated by generating random matrices M and q. All of them turned out to have no feasible solution. Existence of a solution could have been ensured by using results of Mangasarian [12,13]; however a specific solution would not be available. Hence, problems with at least one feasible solution was generated as follows.

- (i) The mxm M-matrix was generated columnwise by generating m^2 uniform random integer variables in the range -50 to +50. The IMSL subroutine GGUBFS was used for this purpose. The elements of the major diagonal, in general, will have some negative elements. In the case this is not so, two columns were interchanged to give at least one negative element. This will ensure that the matrix is not copositive-plus.
- (ii) A number p was arbitrarily set equal to $m/5$. A solution vector \bar{z} was generated with first p elements equal to zero and the remaining (m-p) elements equal to an integer random number between 2 and 5 (inclusive)
- (iii) Partitioning M as

$$M = \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array} \right]$$

Table 1. Results of the Three Indefinite Quadratic Problems (Kough [11])

Problem No.	(# of variables, # of constraints)	# of Lemke pivots	Solution	Computation time in cpu seconds
1	(4,5)	15	x = (12.9924, 1.8515) y = (11.9968, 3.3143)	0.834
2	(4,8)	14	x = (17.4623, 14.7453) * y = (13.5071, 14.0164)	1.029
3	(4,8)	10	x = (16.5911, 0.0) * y = (27.8055, 11.9603)	0.639

* Global optimum

where M_{11} is $p \times p$, the \bar{w} solution vector was generated by

$$\bar{w}_i = \begin{cases} 2 & \text{if } [M_{11}|M_{12}]\bar{z} > 0 \\ 0 & \text{if } [M_{11}|M_{12}]\bar{z} = 0 \\ 1 & \text{if } [M_{11}|M_{12}]\bar{z} < 0 \end{cases} \quad \text{for } i = 1, \dots, p$$

and $\bar{w}_i = 0$ for $i = p+1, \dots, m$.

(iv) The vector q is then uniquely given by $q = \bar{w} - M\bar{z}$.

Note that q_i could be negative (and, in general, it will be). Also note that the solution (\bar{w}, \bar{z}) is complementary.

12. Computational Results

Ten each of 20×20 and 30×30 problems were generated as in Section 3.1 above. They are identified by PB-20-01 to PB-20-10 and PB-30-01 to PB-30-10 respectively in subsequent discussions. The essential characteristics of these problems is given in Table 2.

Table 3 summarizes the computational results. The average time that the algorithm took to solve a problem of size $m = 20$ was 20.259 c.p.u. seconds. For a problem of size 30, two of the problems could not be solved within the limits of 60 allowable cuts for the program. The average computation time for the remaining 8 30×30 problems was 56.16 c.p.u. seconds.

Note that the fathoming procedure was used infrequently, but when used it yielded the solution in all, except one, instance. Also in block 3, no nonextremal variable was identified in all cases.

In column 6 of Table 3 we have presented a measure called Set Difference. If $x = (w, z)$ is the solution at the end of block 1 of the

Table 2. Problem Data

Problem No.	# of Negative Main Diagonal Elements	# of Negative Elements in M Matrix	Density of M	# of Negative Elements in q Vector
PB-20-01	9	202	0.99	11
20-02	11	188	0.9775	12
20-03	11	188	0.9775	15
20-04	8	191	0.97	13
20-05	7	204	0.975	8
20-06	13	200	0.9875	9
20-07	11	184	0.975	14
20-08	10	200	0.975	10
20-09	11	184	0.9750	14
20-10	8	198	0.9825	9
PB-30-01	13	452	0.98	12
30-02	17	436	0.9744	11
30-03	10	421	0.9789	14
30-04	14	440	0.9822	13
30-05	16	428	0.9833	15
30-06	20	442	0.9767	18
30-07	16	460	0.9867	11
30-08	16	439	0.9811	13
30-09	13	438	0.9789	16
30-10	16	460	0.9867	11

Table 3. Summary of Results

Problem No.	Computation time in cpu seconds	Total # of cuts (cuts based on single pairs)	# of times fathoming used	# of nonextremal variables removed	Set Difference S	# of Lemke pivots
PB-20-01	19.76	6(0)	1*	0	16	3
20-02	27.57	8(0)	1*	0	12	8
20-03	26.97	10(3)	1*	0	11	57
20-04	19.44	5(2)	1*	0	8	95
20-05	25.85	6(1)	0	0	6	115
20-06	14.345	2(0)	0	0	9	125
20-07	18.616	6(3)	1*	0	17	9
20-08	12.233	3(1)	0	0	11	54
20-09	17.972	6(3)	1*	0	17	9
20-10	20.016	6(1)	0	0	18	5
Average for 20x20	20.259	5.60	0.60	0	12.5	48.3
PB-30-01	25.21	4(1)	1	0	16	16
30-02	---	---	-	-	-	--
30-03	94.115	19(1)	0	0	19	6
30-04	28.604	4(0)	0	0	15	44
30-05	37.084	6(1)	1*	0	20	50
30-06	113.684	16(2)	0	0	22	64
30-07	61.28	10(0)	0	0	15	27
30-08	32.734	0(0)	0	0	0	376
30-09††	---	---	-	-	--	--
30-10	56.589	10(1)	0	0	13	27
Average for 30x30	56.163	9.86	0.286	0	17.14	76.25

* indicates solution found upon fathoming
†† not solved within program limits (60 cuts)

flow diagram, and $\bar{x} = (\bar{w}, \bar{z})$ is the solution when the algorithm terminated, the set difference S is given by

$$S = \{j: x_j \text{ is basic in } \hat{x} \text{ but not in } \bar{x}\} \cup \{j: x_j \text{ is basic in } \bar{x} \text{ but not in } \hat{x}\}$$

This was computed because it was suspected that an algorithm (like Lemke's) which is likely to yield a solution at the end of block 1 closer to the optimal would have a significant advantage. Table 3 does not bear out this conjecture strongly. Another interesting fact worth mentioning is that in all cases, except Problem PB-30-08, Lemke's algorithm in block 1 ended with a ray termination.

The following problem from [5] was of interest because two columns in the matrix were dependent, the LCP solutions were degenerate and infinite.

$$m = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \cdot q = \begin{pmatrix} 1 \\ -1 \\ 3 \\ 1 \end{pmatrix}$$

An interesting fact about this problem is that the algorithm demonstrated the presence of nonexternal variables and found the solution $w = (2, 0, 0, 0)$ and $z = (0, 0, 4, 1)$.

It may be recalled a measure C defined in (14) of Section 8 was used to drive the algorithm towards a complementary solution. In general C will decrease at each iteration except possibly when a cut is added. In 58 instances of cuts for 20x20 problems, only in 9 instances C increased marginally.

In Section 5 we discussed the alternatives of basing the cuts on a single complementary pair of variables, both positive, versus two pairs of complementary variables. In the latter case, generation of data to set up the four linear subproblems contributed significantly to the total solution time. Table 4 compares these two strategies. The solution times when two pairs were used were on a CDC CYBER 170/730 whereas the solution times when a single pair was used were on a CDC CYBER 170/760 (which is two to three times faster). It clearly shows the superiority of the proposed method to develop deep cuts.

Table 4. Computational Results: Cuts Based on (a) Single Pair
and (b) Two Pairs of Violations

Problem No.	Computation Time in cpu seconds		# of Cuts Added	
	Single Pair†	Two Pairs††	Single Pairs†	Two Pairs††
PB-20-01	10.488	19.760	4	6
20-02	35.260	27.57	17	8
20-03	> 140 secs.	26.97	—	10
20-04	> 180.00	19.44	> 70	5
20-05	16.193	25.85	4	6
20-06	> 140	14.345	—	2
20-07	27.408	18.616	13 cuts	6
20-08	18.857	12.233	10	3
30-02	> 140	> 140	—	—
30-04	45.488	28.604	10	4

† On CDC CYBER 170/760

†† On CDC CYBER 170/730

References

1. Balas, E., "Intersection Cuts from Disjunctive Constraints," Management Science Research Report, No. 330, Carnegie-Mellon University, February 1974.
2. Balas, E., "Disjunctive Programming: Properties of the Convex Hull of Feasible Points," Management Science Research Report 348, GSIA, Carnegie-Mellon University, 1974.
3. Balas, E., "Disjunctive Programming: Cutting Planes from Logical Conditions," Nonlinear Programming, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1975.
4. Balas, E., "Disjunctive Programming," Management Science Research Report No. 415, Carnegie-Mellon University, August 1977.
5. Cottle, R. W. and G. B. Dantzig, "Complementary Pivot Theory of Mathematical Programming," Linear Algebra and Its Applications, 1, 103-125, 1968.
6. Glover, F., "Polyhedral Convexity Cuts and Negative Edge Extensions," Zeitschrift fur Operations Research, 18, pp. 181-186, 1974.
7. Glover, F., "Polyhedral Annexation in Mixed Integer and Combinatorial Programming," Mathematical Programming, Vol. 8, pp. 161-188, 1975. (See also MSRS Report 73-9, University of Colorado, August 1973.)
8. Jeroslow, R. G., "The Principles of Cutting Plane Theory: Part I", (with an addendum), GSIA, Carnegie-Mellon University, February 1974.
9. Jeroslow, R. G., "Cutting Planes for Complementarity Constraints," SIAM Journal of Control and Optimization, Vol. 16, No. 1, pp. 56-62, 1976.
10. Jeroslow, R. G., "Cutting Plane Theory: Disjunctive Methods," Annals of Discrete Mathematics, Vol. 1, pp. 293-330, 1977.
11. Kough, P. F., "The Indefinite Quadratic Programming Problem," Operations Research, 27, 516-533, 1979.
12. Mangasarian, O., "Characterization of Linear Complementary Problems," Mathematical Programming Study, 7, 74-87, 1978.
13. Mangasarian, O., "Characterization of Bounded Solutions of Linear Complementarity Problems," Mathematical Programming Study, 19, 153-166, 1982.
14. Marsten, R. E., Users Guide for XMP, University of Arizona, Tuscon developed under NSF Grant MCS-76-01311 and MCS-76-01311 A01, 1978.
15. Lemke, C. E., "On Complementary Pivot Theory," Mathematics of the Decision Sciences, (G. B. Dantzig and A. F. Veinott, Eds.), 1968.

16. Sherali, H. D. and C. M. Shetty, "Deep Cuts in Disjunctive Programming," Naval Research Logistics Quarterly, Vol. 27, pp. 453-475, 1980.
17. Sherali, H. D. and C. M. Shetty, Optimization with Disjunctive Constraints, Lecture notes in Economics and Mathematical Systems #181, Springer-Verlag, 1980.
18. Sherali, H. D. and C. M. Shetty, "Disjunctive Programming, Polyhedral Annexation Techniques, and Nondominated Disjunctive Cutting Planes," Georgia Institute of Technology, 1979.